WHAT IS CLAIMED IS:

1. A system for managing texture data, the system comprising:

a texture memory controller coupled to a texturing engine; and

a texture memory coupled to the texture memory controller, the texture memory configured to store texture data in a texture map set, the texture map set comprising two or more texture layers, wherein the texture memory controller is configured to access one or more packets from the texture memory and pass the packets to the texturing engine, each packet comprising texture data from one or more texture layers.

2. The system of Claim 1, wherein a packet comprises color values associated with one color, the color values being from two or more texture layers.

3. The system of Claim 2, wherein the packet further comprises alpha values representing the amount of blending of the color values.

4. The system of Claim 1, wherein a packet comprises:

a first red-green-blue texel from a first texture layer; and

a second red-green-blue texel from a second texture layer.

5. The system of Claim 4, wherein the packet further comprises:

a first alpha value associated with the first red-green-blue texel from the first texture layer; and

a second alpha value associated with the second red-green-blue texel from the second texture layer.

6. The system of Claim 4, wherein the packet further comprises:

a first red alpha value, a first green alpha value and a first blue alpha value associated with the first red-green-blue texel from the first texture layer; and

a second red alpha value, a second green alpha value and a second blue alpha value associated with the second red-green-blue texel from the second texture layer.

7. The system of Claim 6, wherein the texturing engine is configured to modify the alpha values for one or more colors without modifying the alpha values for other colors.

8. The system of Claim 1, further comprising a texture cache coupled to the texture memory controller and the texturing engine, the texture cache configured to store packets that have been recently retrieved from the texture memory, the texture cache configured to send recently-retrieved packets to the texturing engine.

9. The system of Claim 8, further comprising a set of texture cache status registers coupled to the texture memory controller and the texture engine, the texture cache status registers configured to store at least one texture attribute associated with the packets stored in the texture cache.

10. The system of Claim 1, wherein the texture memory is configured to store attributes associated with each texture layer, the attributes comprising:

a width of a texture layer;

a height of a texture layer;

a texel format; and

a texel packing factor.

11. The system of Claim 10, wherein the texel format indicates the number of bits per texel, a texel operation mode, and a texture mode flag to indicate whether depth texturing is to be used or not.

12. The system of Claim 10, wherein the texel packing factor is used by the texturing engine to layer several textures together for depth texturing

13. The system of Claim 10, wherein the texel packing factor is used by the texturing engine to layer several textures together for combining several textures in a single rendering pass.

14. The system of Claim 1, wherein the texture memory further comprises an alpha map, the alpha map comprising alpha values, each alpha value represents the amount of blending for a red-green-blue triplet within the texture memory.

15. The system of Claim 1, wherein the texture memory further comprises an alpha map, the alpha map comprising alpha values, each alpha value represents the amount of blending of one color value in a texture layer within the texture memory.

16. A method of managing texture data, the method comprising:

storing a texture map set in a texture memory, the texture map set comprising two or more texture layers.

17. The method of Claim 16, further comprising retrieving one or more packets from the texture memory, each packet comprising texture data from one or more texture layers.

18. The method of Claim 17, further comprising:

receiving a request for one or more packets;

accessing a set of texture cache status registers to determine whether some or all of the requested packets are stored in a texture cache;

passing requested packets stored in the texture cache to a texturing engine if some or all of the requested packets are stored in the texture cache; and

retrieving requested packets from the texture memory if the texture cache does not have all of the requested packets.

19. The method of Claim 17, further comprising retrieving information associated with each texture layer, the information comprising:

a width of a texture layer;

a height of a texture layer;

a texel format; and

a texel packing factor.

20. The method of Claim 19, further comprising storing the information associated with each texture layer in a set of texture cache status registers.

21. The method of Claim 17, wherein a packet comprises color values associated with one color, the color values being from two or more texture layers.

22. The method of Claim 21, wherein the packet further comprises alpha values representing the amount of blending of the color values.

23. The method of Claim 17, wherein a packet comprises:

a first red-green-blue texel from a first texture layer; and

a second red-green-blue texel from a second texture layer.

24. The system of Claim 23, wherein the packet further comprises:

a first alpha value associated with the first red-green-blue texel from the first texture layer; and

a second alpha value associated with the second red-green-blue texel from the second texture layer.

25. The system of Claim 23, wherein the packet further comprises:

a first red alpha value, a first green alpha value and a first blue alpha value associated with the first red-green-blue texel from the first texture layer; and

a second red alpha value, a second green alpha value and a second blue alpha value associated with the second red-green-blue texel from the second texture layer.

26. The method of Claim 25, further comprising modifying the alpha values for one or more colors without modifying the alpha values for other colors.

27. A method of surface depth texturing, the method comprising:

determining a depth stepping angle, the depth stepping angle formed by a first vector from an eye point position to a fragment in a polygon surface and a second vector normal to the surface of the fragment;

converting the depth stepping angle to one or more texel offset values;

applying the texel offset values to a first texel in a first texture layer to find a corresponding second texel in a second texture layer;

blending one or more color values and one or more alpha values associated with the first and second texels to create an apparent surface depth for the fragment; and

applying the blended color values and alpha values to the fragment.

28. The method of Claim 27, wherein converting the depth stepping angle to one or more texel offset values comprises adjusting an amount of parallax between the first and second texture layers.

29. The method of Claim 28, wherein adjusting an amount of parallax comprises left-shifting a set of binary bits representing the texel offset values.

30. The method of Claim 27, further comprising right-shifting a set of binary bits representing the texel offset values by one bit for a selected second level MIP map, by two bits for a selected third level MIP map, by three bits for a selected fourth level MIP map.

31. The method of Claim 27, wherein the texel offset values are positive.

32. The method of Claim 27, wherein the texel offset values are negative

33. The method of Claim 27, wherein the color values comprise red, green and blue.

34. The method of Claim 27, wherein the alpha values comprise a red alpha value, a green alpha value and a blue alpha value.

35. The method of Claim 34, further comprising modifying the alpha values for one or more colors without modifying the alpha values for other colors.

36. The method of Claim 27, wherein the alpha values from the second texel control blending of the first and second texel color values with an interpolated polygon color at a location of the fragment.

37. The method of Claim 27, wherein the acts are repeated for a second fragment with a second depth stepping angle.

38. The method of Claim 27, further comprising:

caching the second texel and a third texel; and

accessing a fourth texel from a texture memory and the third texel from a texture cache when the acts are repeated for a second fragment with a second depth stepping angle.

39. The method of Claim 27, wherein the acts are preformed in real-time.

40. A surface-depth texturing system comprising:

a texturing engine; and

a texturing memory coupled to the texturing engine, wherein the texturing engine is configured to retrieve a first texel in a first texture layer from the texture memory; determine a depth stepping angle between a first vector from an eye point position to a fragment in a polygon surface and a second vector normal to the surface of the fragment; convert the depth stepping angle to one or more texel offset values; apply the texel offset values to the first texel to find a corresponding second texel in a second texture layer; blend one or more color values and one or more alpha values associated with the first and second texels to create an apparent surface depth for the fragment; and apply the blended color values and alpha values to the fragment.

41. The texturing system of Claim 4, wherein each texel comprises a red color value, a red alpha value, a green color value, a green alpha value, a blue color value and a blue alpha value.